



# knowledge\_transformer



# 1.

1. YAML / API
  2. API appid/key
  3. HTTP
  - 4.
  5. Tracing ID Prometheus Flower
  6. Celery Webhook Result Backend

## 2.

Office REST API Celery FastAPI task\_id Celery Worker LibreOffice / Inkscape / FFmpeg Result Backend Webhook Celery Pipeline Celery Chain/Group/Chord

### 3.

- sequenceDiagram participant API as API participant Q as Redis participant W as Celery Worker participant P as participant S as participant CB as API->>API:

- ```

    task_id API->>Q: conversion.handle_batch Q-->>W: W-
    >>S: URL W->>P: P-->>W: W-
    >>S: converted/task_id/ W->>API: results[] alt
    callback_url API->>CB: Webhook else API-->>API: end
      i. task_id
      ii. conversion.handle_batch Redis
      iii. Worker / URL
      iv. converted/{task_id}/
      v. results[] callback_url

```
- sequenceDiagram participant C as participant API as API
 participant R as Redis / Result Backend
 participant W as Celery Worker
 participant M as Mon as Flower/Prometheus
 C->>API: POST /api/v1/convert API-->>C: 202 Accepted + task\_id API-
 >>R: conversion.handle\_batch R->>W: W->>M: / W-->>R: alt callback\_url W->>C: Webhook
 else C->>R: C->>M: end Mon-->>API: Mon-->>W:
  - Pipeline Celery /api/v1/convert API Redis
  - API HTTP 202 task\_id ID
  - Celery Worker Webhook Result Backend Webhook Result Backend
  - Flower Prometheus API/Worker

## 4. Celery

- Celery
  - priority prefetch\_multiplier worker task\_time\_limit soft\_time\_limit acks\_late

- API files[] SLA priority conversion.handle\_batch Worker max\_tasks\_per\_child batch\_size

◆◆◆◆

```

function submit_request(files, metadata):
    priority = calc_priority(files, metadata.sla)
    batch = split_files(files, metadata.batch_size)
    for chunk in batch:
        payload = build_payload(chunk, priority, metadata)
        push_to_queue(select_queue(chunk.plugin), payload)

worker_loop():
    settings = load_runtime_limits()
    while worker_alive():
        task = fetch_from_queue(settings.prefetch)
        if not task:
            continue
        with deadline(settings.task_time_limit):
            try:
                for file in task.files:
                    artifact = run_plugin(file)
                    upload_to_storage(task.task_id, artifact)
                    ack(task)
            except TimeoutError:
                mark_failed(task, "timeout")
                requeue_remaining(task)
            except Exception as exc:
                mark_failed(task, str(exc))
                maybe_retry(task)

```

• ◆◆◆◆◆

- LibreOffice headless soffice Office→PDF/HTML/ API URL source\_format filter



- x264 Baseline + CRF 23  
AAC 128kbps
- b. wav→mp3 16kHz/H.265 → H.264
  - c. mov→mp4 H.265 → H.264
  - d. gif→mp4/png + scale MP4 → PNG

## 5. lib

|                                     |                                  |                                                                                                                      |  |
|-------------------------------------|----------------------------------|----------------------------------------------------------------------------------------------------------------------|--|
| rag_converter.plugins               | ConversionInput/ConversionResult | source_format → target_format → input_path / input_url / object_key → metadata → output_path → object_key → metadata |  |
|                                     |                                  |                                                                                                                      |  |
| ConversionPlugin.convert(payload)   |                                  | payload: ConversionInput                                                                                             |  |
| ConversionPlugin.describe()         |                                  |                                                                                                                      |  |
| PluginRegistry.register(plugin_cls) |                                  | plugin_cls: Type[ConversionPlugin]                                                                                   |  |
| PluginRegistry.get(source, target)  |                                  | source:str, target:str                                                                                               |  |
| PluginRegistry.list()               |                                  |                                                                                                                      |  |
| load_plugins(module_names=None)     |                                  | module_names: Iterable[str]                                                                                          |  |
| read_plugin_module_file(path)       |                                  | path: str                                                                                                            |  |

|                                                                                                                                                                                   |                                                 |                             |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|-----------------------------|
|                                                                                                                                                                                   | ◇◇/◇◇                                           |                             |
|                                                                                                                                                                                   | write_plugin_module_file(path, modules)         | `path: str                  |
| <ul style="list-style-type: none"> <li>scripts/manage_plugins.sh ◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇</li> <li>rag_converter.monitoring ◇◇ API/Worker ◇◇ Prometheus ◇◇</li> </ul>                     |                                                 |                             |
|                                                                                                                                                                                   | ◇◇                                              | ◇◇◇◇                        |
|                                                                                                                                                                                   | ensure_metrics_server(port)                     | port:int Prometheus ◇◇◇     |
|                                                                                                                                                                                   | record_task_accepted(priority)                  | priority:str                |
|                                                                                                                                                                                   | record_task_completed(status)                   | status:str                  |
|                                                                                                                                                                                   | collect_dependency_status(settings, celery_app) | settings:Settings ◇ celery_ |
|                                                                                                                                                                                   | _check_redis(settings)                          | settings:Settings           |
|                                                                                                                                                                                   | _check_minio(settings)                          | settings:Settings           |
|                                                                                                                                                                                   | _check_celery_workers(celery_app)               | celery_app:Celery           |
| <ul style="list-style-type: none"> <li>rag_converter.logging ◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇ JSON◆Trace/TaskID ◆<br/>FastAPI/Celery ◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇◇ stdout ◇◇◇◇◇◇◇◇ ◆<br/>Loki◆ELK◆◆</li> </ul> |                                                 |                             |

|                                                                |                                           |                             |
|----------------------------------------------------------------|-------------------------------------------|-----------------------------|
| configure_logging(settings)                                    | settings:LoggingSettings<br>level:log_dir | logging + structlog Handler |
| pipeline_service                                               | Celery Pipeline                           |                             |
| CeleryConverterClient.submit_conversion_chain(files, priority) | files>List[                               |                             |
| submit_conversion_group(file_groups, priority)                 | file_groups                               |                             |
| submit_conversion_chord(file_batches, priority)                | file_batches                              |                             |
| get_result(task_id, timeout)                                   | task_id:str                               |                             |
| check_status(task_id)                                          | task_id:str                               |                             |
| quality_check_task(conversion_results)                         | conversion_results                        |                             |
| post_process_task(checked_results)                             | checked_results                           |                             |
| aggregate_results_task(batch_results)                          | batch_results                             |                             |

| ◆◆/◆◆                                | ◆◆                                                                                                                                                                                      |
|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| POST /api/v1/convert                 | Headers: X-Appid , X-Key ◆Body: task_name , priority , call<br>storage{endpoint,access_key,secret_key,bucket} ◆◆◆◆◆ fi<br>target_format , input_url / object_key / base64_data ◆◆◆◆◆ fi |
| GET /api/v1/formats                  | Headers: X-Appid , X-Key                                                                                                                                                                |
| GET /api/v1/monitor/health           | Headers: X-Appid , X-Key                                                                                                                                                                |
| GET /healthz                         | ◆                                                                                                                                                                                       |
| Celery ◆◆<br>conversion.handle_batch | payload = {task_id, files[], priority, callback_url, storag                                                                                                                             |
| Pipeline<br>submit_conversion_chain  | files[] , priority                                                                                                                                                                      |
| Pipeline<br>submit_conversion_group  | file_groups[][] , priority                                                                                                                                                              |
| Pipeline<br>submit_conversion_chord  | file_batches[][] , priority                                                                                                                                                             |

7.    ?    ?    ?    ?

## 8. 🔮

- Celery Chain 🔮
- appid/key 🔮
- Redis 🔮 LibreOffice 🔮 Inkscape 🔮 FFmpeg 🔮
- config/settings.yaml 🔮

## 9. 🔮

| Input                           | Output                | Tool                     | Result             |
|---------------------------------|-----------------------|--------------------------|--------------------|
| doc, docx,<br>ppt, xls          | docx,<br>pdf,<br>html | LibreOffice<br>soffice 🔮 | 🔮                  |
| svg, eps, pdf                   | png,<br>jpeg,<br>webp | Inkscape CLI             | 🔮/🔮                |
| gif, webp                       | png,<br>mp4           | GIF/WebP<br>+ FFmpeg     | 🔮                  |
| wav, flac,<br>ogg, aac          | mp3                   | FFmpeg<br>Audio 🔮        | 🔮                  |
| avi, mov,<br>mkv, webm,<br>mpeg | mp4                   | FFmpeg<br>Video 🔮        | 🔮                  |
| ?                               | ?                     | ?                        | CAD→PDF 🔮 AI→SVG 🔮 |

## 10. 🔮

- Worker 🔮 /tmp/rag\_converter/<task> 🔮
- converted/{task\_id}/... 🔮 bucket 🔮

**11.**



| ?               | ?                                                                       | ?                                                                |
|-----------------|-------------------------------------------------------------------------|------------------------------------------------------------------|
| file_limits     | default_max_size_mb ,<br>per_format_max_size_mb ,<br>max_files_per_task | ?                                                                |
| logging         | level , log_dir ,<br>max_log_file_size_mb ,<br>backup_count             | ?                                                                |
| monitoring      | prometheus_port ,<br>metrics_interval_sec ,<br>health_api ◊             | ?                                                                |
| minio           | endpoint , access_key ,<br>secret_key , bucket ,<br>timeout             | http://localhost:9000 /<br>minioadmin / minioadmin /<br>qadata ◊ |
| convert_formats | source , target ,<br>plugin                                             | ?                                                                |
| api_auth        | required ,<br>app_secrets_path ,                                        | API ?                                                            |

|            |                                                                                  |                      |
|------------|----------------------------------------------------------------------------------|----------------------|
| ◆◆◆        | ◆◆◆◆                                                                             | ◆◆                   |
|            | header_appid                                                                     |                      |
| celery     | broker_url ,<br>result_backend ,<br>task_time_limit_sec ,<br>prefetch_multiplier | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆     |
| rate_limit | enabled , interval_sec ,<br>max_requests                                         | API ◆◆◆◆◆            |
| ◆◆         | service_name ,<br>plugin_modules_file ◆                                          | ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆ |

- `RAG_YAML` `RAG_REDIS_URL` `RAG_MINIO_ENDPOINT` `RAG_API_AUTH_REQUIRED`

| ?                       | ?                                | ? | ? | ?                 | ?              | ?      | ? |
|-------------------------|----------------------------------|---|---|-------------------|----------------|--------|---|
| RAG_REDIS_URL           | celery.broker_url/result_backend | ? | ? | Celery Broker     | Result Backend | ?      | ? |
| RAG_MINIO_ENDPOINT      | minio.endpoint                   | ? | ? | ?                 | ?              | Docker | ? |
| RAG_API_AUTH_REQUIRED   | api_auth.required                | ? | ? | ?                 | ?              | API    | ? |
| RAG_FILE_LIMIT_MAX_SIZE | file_limits.default_max_size_mb  | ? | ? | ?                 | ?              | ?      | ? |
| RAG_PROM_PORT           | monitoring.prometheus_port       | ? | ? | Prometheus        | ?              | ?      | ? |
| ?? RAG_*                | ??→??                            | ? | ? | pydantic-settings | ?              | ?      | ? |
|                         |                                  | ? | ? | Kubernetes        | ?              | ?      | ? |
|                         |                                  | ? | ? | Compose           | ?              | ?      | ? |

- CLI/`manage_plugins.sh` `make_key.sh`

|                                        |                                                                                 |                                                          |
|----------------------------------------|---------------------------------------------------------------------------------|----------------------------------------------------------|
| <code>manage_plugins.sh</code>         | <code>config/plugins.yaml</code>                                                | <code>list/install/remove</code>                         |
| <code>make_key.sh</code>               | <code>API</code><br><code>appid/key</code><br><code>secrets/appkeys.json</code> | <code>--appid</code> <code>ID</code>                     |
| <code>docker-start.sh / stop.sh</code> | <code>Docker Compose</code>                                                     | <code>.env</code><br><code>docker compose up/down</code> |
| <code>show_server.sh</code>            | <code>API/Worker/Redis</code>                                                   |                                                          |

## 12.

- `start_server.sh` FastAPI Celery Worker Flower

|                            |                                                                                     |
|----------------------------|-------------------------------------------------------------------------------------|
| <code>FastAPI</code>       | <code>uvicorn rag_converter.app:app --host 0.0.0.0 --port \${API_PORT}</code>       |
| <code>Celery Worker</code> | <code>celery -A rag_converter.celery_app.celery_app worker -l \${CELERY_LOG}</code> |
| <code>Flower</code>        | <code>celery -A rag_converter.celery_app.celery_app flower --port \${FLOWER}</code> |

- Celery Worker Pipeline Celery `.env`

`CELERY_BROKER_URL` `CELERY_RESULT_BACKEND` `MINIO_*`

| ?                  | ?                                                                                                                                      | ?                   |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| ?                  | CELERY_BROKER_URL ,<br>CELERY_RESULT_BACKEND ,<br>CELERY_DEFAULT_QUEUE ,<br>CELERYD_PREFETCH_MULTIPLIER ,<br>TASK_TIME_LIMIT , MINIO_* | Broker/Backend<br>? |
| Pipeline<br>Celery | BROKER_URL , RESULT_BACKEND ,<br>MINIO_ENDPOINT ,<br>MINIO_ACCESS_KEY ,<br>MINIO_SECRET_KEY ,<br>PIPELINE_QUEUE                        | Redis<br>?          |

# 13.

| ❖❖          | ❖❖❖❖/❖❖        | ❖❖                               |
|-------------|----------------|----------------------------------|
| FastAPI     | ≥ 0.104        | ❖❖ REST API ❖❖❖❖                 |
| Celery      | ≥ 5.3          | ❖❖❖❖❖❖❖❖❖                        |
| Redis       | ≥ 7            | ❖❖ Celery Broker/Result Backend❖ |
| ❖❖❖❖        | S3 ❖❖❖❖ MinIO❖ | ❖❖❖❖/❖❖❖❖❖                       |
| LibreOffice | ❖❖ LTS         | ❖❖❖❖❖❖❖                          |
| Inkscape    | ≥ 1.3          | ❖❖❖❖❖❖❖❖                         |
| FFmpeg      | ≥ 5.0          | ❖❖❖❖❖/❖❖❖                        |
| Flower      | ≥ 1.2          | Celery ❖❖ UI❖                    |
| Prometheus  | ≥ 2.46         | ❖❖❖❖❖❖❖❖                         |

- Python `pyproject.toml` Pipeline `requirements.txt`   
  README

## 14. 🔮🔮🔮🔮🔮

- pytest + HTTPX Celery test worker mock 🔮🔮🔮
- /Pipeline Chain/Group/Chord 🔮🔮
- Pipeline Chain/Group/Chord 🔮
- Mock 🔮

| File                    | Format       | Format → Format    | Size (MB) | Description                     |
|-------------------------|--------------|--------------------|-----------|---------------------------------|
| doc_sample_small.doc    | doc          | doc → docx         | 0.05      | Chain 🔮                         |
| doc_sample_pdf.doc      | doc          | doc → pdf          | 0.05      | PDF 🔮 sof                       |
| html_inline_base64.json | html(base64) | html(base64) → pdf | 0.01      | base64_d 🔮                      |
| ppt_marketing.ppt       | ppt          | ppt → pdf          | 48        | base64_d 🔮                      |
| svg_logo.svg            | svg          | svg → png          | 0.8       | /Pipeline Chord 🔮               |
| gif_banner.gif          | gif          | gif → mp4          | 25        | base64_d 🔮                      |
| webp_large.webp         | webp         | webp → png         | 15        | base64_d 🔮                      |
| wav_podcast.wav         | wav          | wav → mp3          | 180       | base64_d 🔮                      |
| flac_archive.flac       | flac         | flac → mp3         | 220       | FILE_TO                         |
| mov_trailer.mov         | mov          | mov → mp4          | 480       | base64_d 🔮 500 Pipeline Chord 🔮 |
| mkv_fail.mkv            | mkv          | mkv → mp4          | 300       | base64_d 🔮                      |
| invalid_format.bin      | bin          | bin → docx         | 1         | /base64_d 🔮                     |
| auth_test.docx          | docx         | docx → pdf         | 2         | API key 🔮                       |
| webhook_payload.json    | -            | -                  | -         | Webhook 🔮 🔮 🔮 🔮 🔮               |

- Mock 🔮

| ???   | ???      | ???                          | <b>Mock</b> ???                                                      |
|-------|----------|------------------------------|----------------------------------------------------------------------|
| UT-01 | API ??   | API ????                     | Mock Redis + Celery ?? HTTPX client                                  |
| UT-02 | API ??   | ???                          | Mock <code>file_limits</code> ??                                     |
| UT-03 | Worker   | ???                          | Mock MinIO SDK ?? <code>S3Error</code>                               |
| UT-04 | Worker   | ???                          | Mock ?? <code>convert</code> ?? <code>/tmp/out.docx</code>           |
| UT-05 | Worker   | ???                          | Mock Celery ??                                                       |
| UT-06 | ???      | API Key ??                   | Mock <code>make_key.sh</code> +<br><code>secrets/appkeys.json</code> |
| UT-07 | ???      | Webhook ??/<br>???           | Mock <code>requests.post</code>                                      |
| UT-08 | Pipeline | Chain ???                    | Mock<br><code>conversion/quality_check/post_process</code>           |
| UT-09 | Pipeline | Group +<br>Result<br>Backend | Mock <code>AsyncResult</code>                                        |
| UT-10 | ???      | ???                          | Mock Redis/MinIO/Celery Ping                                         |
| UT-11 | ???      | Logging ??                   | Mock <code>structlog.configure</code>                                |
| UT-12 | Worker   | base64<br>???                | Mock/?? MinIO                                                        |
| UT-13 | Worker   | ???                          | Mock MinIO ???                                                       |

- ??/??/???
  - ?? UT-01/UT-07 ?? pytest-xdist +  
Celery test worker ?? `queue_depth` ?? `prefetch_multiplier`

| ?       | ?                                    | ?                                          |
|---------|--------------------------------------|--------------------------------------------|
| TC-0001 | doc_sample_small.doc , priority=high | HTTP 202 ? task_id ? ? ? ? ? ?             |
| TC-0002 | flac_archive.flac (220MB)            | HTTP 400 ? ? ? ? FILE_TOO_LARGE            |
| TC-0003 | wav_podcast.wav , Webhook=200        | results[].status=success ? Webhook success |
| TC-0004 | mkv_fail.mkv + ? ? ? ?               | ? ? ? ? ? ? reason=timeout ? ? ? ? ?       |
| TC-0005 | Pipeline Chain ? ? ? ? ?             | stage=completed ? quality_score>0.9        |
| TC-0006 | API Key ? ? / ? ?                    | ? key ? ? ? ? key ? ? 401                  |

A horizontal row of eight empty diamond-shaped boxes, each containing a question mark, used for a matching exercise.

| ◇◇?                      | HTTP<br>◇◇ | ◇??  | ◇?                      | ◇?◇?                                     |
|--------------------------|------------|------|-------------------------|------------------------------------------|
| ERR_AUTH_INVALID         | 401        | 4011 | ◇?◇?◇?appid<br>◇ key ◇? | ◇?◇?◇?key ◇?<br>key?                     |
| ERR_FILE_TOO_LARGE       | 400        | 4201 | ◇?◇?◇?◇?                | ◇?◇<br>per_format_max_<br>flac_archive.f |
| ERR_BATCH_LIMIT_EXCEEDED | 400        | 4202 | ◇?◇?◇?/<br>◇?◇?◇?       | ◇?◇ max_files_/<br>max_total_upload      |
| ERR_FORMAT_UNSUPPORTED   | 400        | 4203 | ◇?◇?◇?◇?                | ◇?◇?◇?◇?◇?<br>invalid_format             |
| ERR_TASK_FAILED          | 500        | 5001 | ◇?◇?◇?◇?                | ◇?◇?◇?◇?◇?<br>◇?◇?◇?◇?◇?◇?               |